Signal Processing 110 (2015) 155–163

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast detection of human using differential evolution ☆

Ni Chen [a,c,d], Wei-Neng Chen [b,c,d,*], Jun Zhang [b,c,d]

[a] Department of Computer Science, Sun Yat-Sen University, Guangzhou, China
[b] School of Advanced Computing, Sun Yat-Sen University, Guangzhou, China
[c] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[d] Engineering Research Center of Supercomputing Engineering Software, Ministry of Education, China

ARTICLE INFO

Article history:
Received 3 May 2014
Received in revised form
13 August 2014
Accepted 29 August 2014
Available online 6 September 2014

Keywords:
Human detection
Histograms of oriented gradients
Differential evolution

ABSTRACT

Human detection is a significant and challenging task with applications in various domains. In real-time systems, the speed of detection is crucial to the performance of system, while the accuracy is also taken into consideration. In this work, a human detection approach based on Histograms of Oriented Gradients (HOG) feature and differential evolution (DE), termed as HOG-SVM-DE, is proposed to achieve both fast and accurate detection. The proposed method considers the problem of locating an objective detection window as a search problem, and speeds up the detection stage by solving the search problem with DE. DE is chosen as the optimizer as it is characterized by fast and global convergence. The proposed system trains only one linear-SVM, and allows tradeoffs between the detection rate and the detection time to satisfy different applications by simply tuning one parameter. Experiments are conducted on a set of images from the INRIA Person Dataset, and the results validate that the proposed HOG-SVM-DE is promising in terms of both speed and accuracy.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Human detection is a significant and challenging task in computer vision. With the increased popularity of computer vision in areas such as video surveillance, smart vehicles, robotics, and ubiquitous system, technologies of human detection have drawn much research attention.

In many real-time applications, the detection of human is required to be both fast and accurate. The task is difficult since pedestrians are usually with various appearances, postures and backgrounds. There are mainly two directions for the development on human detection, i.e. feature and classifier. Methods such as Haar Wavelet features [1], Implicit Shape Models [2], edge templates [3], Adaptive Contour Features [4,5], and Histogram of Oriented Gradients (HOG) [6] have been proposed for extracting features. On the other hand, classifiers for human detection mainly include Support Vector Machines (SVM) [6–9] and cascade-structured boosting-based classifiers [10,11]. For methods that adopt SVM, there is a preference for linear SVM because it achieves higher speed and reduces the problem of overfitting compared with the non-linear SVM kernels [6].

Among the feature extraction methods, the Histograms of Oriented Gradients (HOG) proposed by Dalal and Triggs [6] is widely used for human detection for its robustness and fast speed. Working together with linear SVM, the HOG features are extracted in sliding window fashion for human detection in an image. The HOG-SVM approach scans the whole image

☆ This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China No. 2013AA01A212, in part by the NSFC for Distinguished Young Scholars 61125205, in part by the NSFC Nos. 61379061, 61070004, and U1201258.
* Corresponding author at: School of Advanced Computing, Sun Yat-Sen University, Guangzhou, China.
E-mail address: chenwn3@mail.sysu.edu.cn (W.-N. Chen).

http://dx.doi.org/10.1016/j.sigpro.2014.08.044
0165-1684/© 2014 Elsevier B.V. All rights reserved.

in a sliding window in order to locate the objective detection windows. However, the time efficiency of the scanning strategy hardly satisfies the requirement of real-time applications for although the HOG feature extraction is faster than many counterparts. Later on, Zhu et al. [12] speed up the method by constructing cascade-of-rejectors with AdaBoost algorithm. Although their method performed favorably in terms of detection time, the training process is much time-consuming. An et al. [13] suggested using a particle swarm optimizer (PSO) for locating the detection window, but the enhanced speed is at the cost of accuracy. To compute the HOG features more efficiently, in the work of Pang et al. [14], the HOG-SVM based human detection was accelerated by reusing the features in blocks and cell-based interpolation. Furthermore, the Dominant Orientation Templates (DOT) [15,16] which is based on the idea of HOG is proposed to enhance the speed.

In this work, a human detection approach termed HOG-SVM with differential evolution (HOG-SVM-DE) is proposed. Aiming at both fast detection and training cost minimization, the proposed method focuses on locating an objective detection window quickly and accurately at real-time. To fulfill the task, the evolutionary computation technology differential evolution (DE) is applied instead of scanning the detection windows in sliding fashion. The DE is a simple and efficient population-based search algorithm proposed by Storn and Price [17,18] for global optimization. Characterized by its fast and global convergence, the algorithm has been successfully applied to various real-world applications in diverse domains [19,20]. In our proposed method, the DE algorithm works at the real-time detection stage and solves the non-linear search problem of locating the objective detection window. Individuals in DE are defined as detection windows on the image, and the fitness of individuals is assigned according to the output of HOG-SVM for the corresponding detection window.

The proposed method has following features:

- The proposed HOG-SVM-DE provides an approach for fast detection at real time, and at the same time requires minimum off-line training. Only a single linear SVM is required to be trained.
- A human detection framework based on solving the human detection problem as a search problem is suggested. In the proposed method, the feature extraction method and classifier are relatively independent of the search algorithm. Although HOG feature and linear SVM are adopted in this work, other developments of both feature extraction approaches and classifiers might be applied to the proposed framework.
  Examples are the simple graph-based method proposed in [21], the adaptive hypergraph learning method proposed in [22], and the multimodal features and classifiers like the multiview sparse learning methods [23] and the high-order multiview distance learning based classifier [24]. Besides, the information-reusing strategy in [14] can be utilized in the proposed framework to accelerate the computing of HOG features.
- The proposed algorithm is tunable, which means that the balance between the successful rate and the execution time can be adjusted by simply changing a parameter. In this way, the method can satisfy the requirement of different applications with little modification.

The remainder of this paper is organized as follows. Section 2 introduces the framework of HOG-SVM based human detection and the search problem defined in the framework. Section 3 describes the proposed HOG-SVM-DE in detail. In Section 4, the proposed algorithm is simulated in a set of experiments and the results are discussed. Finally, Section 5 concludes the paper.

2. HOG-SVM based human detection as a search problem

Characterized by relatively fast computation and favorable performance, the HOG-based human detection and its variants are widely used for human detection. A typical framework of the method is presented in Fig. 1, where the three modules of the system, i.e. HOG feature extraction, classifier training, and detection module are illustrated. This framework is explained as follows.

There are two stages in the human detection system in Fig. 1, i.e. off-line training stage and online detection stage. The training stage works only once, whereas the online detection stage works every time when a real-time image arrives for detection.

The procedure of HOG feature extraction is employed in both the training stage and the detection stage. Proposed by Dalal and Triggs [6], the technique extracts a feature set from a detection window. The extraction process is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid.

In the off-line training stage, a classifier, typically linear SVM, is trained for human/non-human classification on a single detection window. For the training of classifier, a number of labelled sample images are fed to the HOG feature extraction module. The HOG features are then extracted and used to train the classifier.

In the online detection stage, real-time images arrive for human detection. The detection module is responsible for finding out the objective windows containing human

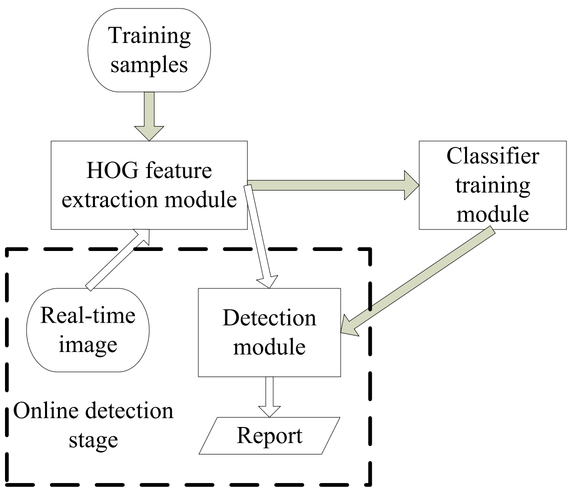

Fig. 1. Architecture of HOG-SVM based human detection.

image from numerous possible detection windows. One way of locating the objective window is scanning the whole image in a sliding fashion. In this process, to judge whether or not a window is the objective window, the HOG feature of the window is extracted and fed to the classifier trained in the off-line training stage.

In the detection stage introduced above, finding the objective windows from possible detection windows can be viewed as a time-consuming search problem. There are numerous possible detection windows in an image with different positions and sizes, and judging a window means calling the HOG-SVM procedure. Thus searching for the objective windows requires much computational effort.

The problem of finding the objective detection window containing human image can be formulated as follows. For a detection window (px, py, $winsize$) on an image, px and py define the positions, i.e. the distances from the upper left of detection window to the left edge and the top edge of the image, respectively. The $winsize$ represents the size of detection window.

In the detection stage, after the extraction of HOG features on detection window (px, py, $winsize$) and feeding the features to the trained SVM classifier, the output of the SVM classifier is presented in

$$f_{Output} = HOG\_SVM(px, py, winsize) - \theta \quad (1)$$

where the constant θ is a pre-defined threshold, which is usually set to 0. It is expected that a detection window with positive value of f_{Output} rightly contains a human. Besides, a detection windows with larger value of f_{Output} is more likely to be the objective window.

In Fig. 2 the landscape of the HOG-SVM output function in (1) on a sample image is presented, with each position on the contour graph representing a detection window, and the color representing different values of f_{Output}. Here px and py are corresponding to the x- and y-axis, respectively, whereas $winsize$ is fixed to be the standard size for simplicity. It is observed that the area with maximum value of f_{Output} on the function landscape is corresponding to the region containing a human object on the image, which is the same as our expectation.

Given Eq. (1) and Fig. 2, it can be concluded that finding the detection window containing a human object can be converted to the problem of finding a detection window (px, py, $winsize$) satisfying $f_{Output} > 0$. In the case of single human detection, a detection window that maximizes the value of f_{Output} would be preferred.

In this work, we would focus on the online detection module by solving the search problem with consideration of time efficiency.

3. HOG-SVM-DE

3.1. Overall method

In this work, the proposed architecture of HOG-SVM-based human detection is similar to Fig. 1, with an emphasis on the online detection module. As explained in Section 2, the online detection module of the HOG-SVM-based human detection system can be viewed as a search problem. Since computing the objective function, i.e. HOG-SVM output function, is the most time-consuming part of the module, the efficiency of the search algorithm would be crucial in real-time applications that demand fast detection.

For fast detection based on HOG-SVM, our method adopts the DE algorithm to solve the search problem in the online detection module. DE is a population-based meta-heuristic search algorithm for global optimization [17,18]. Similar to other population-based optimization algorithms like particle swarm optimization [25] and ant colony optimization [26], DE approximates the global best solution of a search problem by executing an evolutionary process iteratively. Characterized by its fast and global convergence, the DE algorithms have been applied in many fields for their fast and global convergence on many optimization problems.

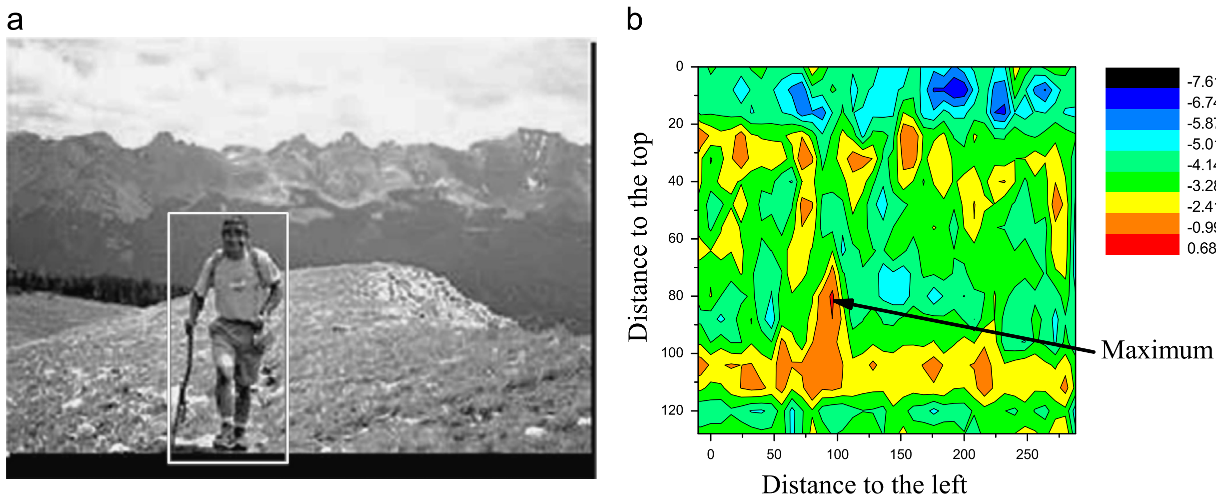

Fig. 2. An example of the landscape of the HOG-SVM output function. (a) Original image. The rectangle containing human image is in the same position as the point with maximum value of HOG-SVM output function in (b). (b) Contour graph for the landscape of HOG-SVM output function for the image in (a). The parameter winsize in (1) is fixed to be the standard size to make the landscape illustratable. The region with maximum function value is marked on the graph.

Instead of traversing the whole search space, the DE algorithm maintains a population of 3-dimensional vectors representing detection windows in the search space. In each generation, the population reproduces new solutions by learning from the difference of other individuals. When a new detection window is generated, the result of HOG-SVM output function is computed and assigned to the individual as fitness value. In each generation, detection windows with better fitness values are more likely to survive. After several generations, the population would be evolved and the best-so-far solution would be reported.

The overall method of the proposed detection system is summarized in Table 1. Details of the proposed method would be explained in the rest of this section.

3.2. Feature extraction

For a detection window from an image, the following steps are taken to obtain the HOG feature:

(1) A simple 1-D $[-1, 0, 1]$ gradient filter is applied to the detection window. Specifically, if the detection window is not the standard size of 64×128, bilinear interpolation is used to resize the image before the gradient filter. The gradient vectors for color images are obtained by computing separate gradients for each channel and taking the one with largest L2-norm.
(2) The window is divided into cells of 8×8 pixels, and each group of 2×2 cells are considered as a block. Blocks are in a sliding fashion. Over each cell, the 9-bin histogram for edge orientation is constructed by each pixel voting for the histogram. In this way, there are 4 cells of 9 bins histogram in each block.
(3) The HOG feature vector consists of histograms extracted from $7 \times 15 = 105$ blocks and each block is a 36-dimensional feature vector in a sliding fashion. To obtain the overall feature set of the window, the feature vector for each block is normalized using L2-norm. Totally, the final feature set would be a vector of $7 \times 15 \times 4 \times 9 = 3780$ features.

The flowchart of HOG feature extraction is summarized in Fig. 3. The detailed implementation of HOG can be referred to [6].

3.3. Search algorithm

In this work, DE is adopted as the search algorithm in the online detection module. The DE algorithms search iteratively for a global optimum with a population of individuals, each represented by a vector $x = [x^g_{i,1}, x^g_{i,2}, \ldots, x^g_{i,D}]$, $i = 1, 2, \ldots, popsize$, where D denotes the dimension of the search space, g is the number of iteration, and *popsize* is the size of population.

The basic flow of DE is illustrated in Fig. 4. After the initialization of population, in each iteration the individuals of DE undergo an evolutionary process which consists of the reproduction and selection operators. In the reproduction operation, the mutation operator produces mutants by learning from the differences between individuals, and the crossover operator combines mutants with original individuals. In the selection operation, the reproduced individuals with better values of fitness function are more likely to survive to the next iteration. The algorithm terminates when a pre-defined criterion is satisfied, e.g. the maximum number of function evaluations is reached.

The details of applying DE to the detection module would be explained in the rest of this section.

3.3.1. Coding of solution

Each individual in the proposed algorithm defines a detection window. For the i-th individual in the g-th generation in DE, the solution vector (x_1, x_2, x_3) is corresponding to the triplet (*px*, *py*, *winsize*) introduced in Section 2. Here *px* and *py* define the positions of window on the image, and *winsize* defines the size of window. In the proposed method, on an image with a size $W \times H$, a window in the position (X, Y) and with a size $Hw/2 \times Hw$ is coded into $[px, py, winsize]$, where $px = X/W, py = Y/W$

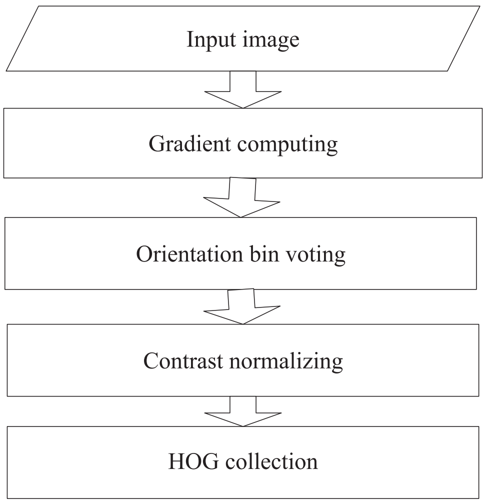

Fig. 3. Basic flow of the HOG feature extraction.

Table 1
Basic flow of HOG-SVM-DE human detection system.

Stage	Steps
Training stage	Step 1: Extract HOG features from positive and negative training samples. Step 2: Train a linear SVM with extracted features.
Detection stage	Step 1: Input real-time image for detection. Step 2: Search for the objective detection window with DE. The SVM trained in training stage is used for the judging whether a window is an objective window. Step 3: Report the results.

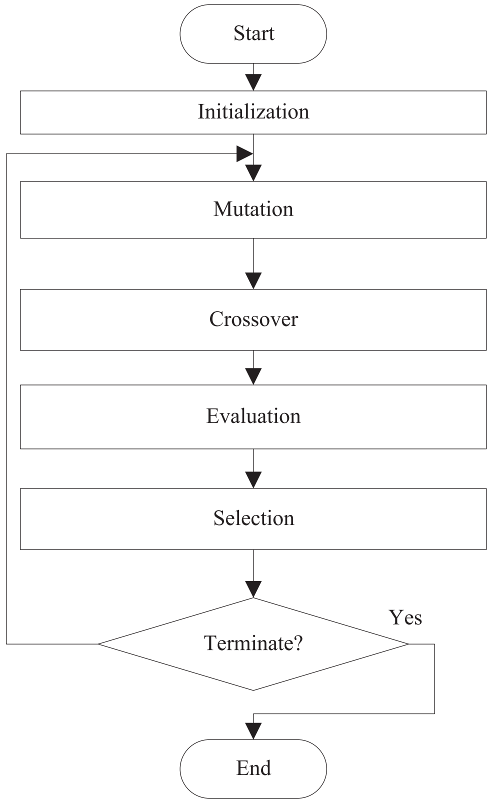

Fig. 4. Flowchart of DE.

and $winsize = Hw/128$. Note that both px and py are normalized to be within the range $[0, 1]$. The parameter $winsize$ can be explained as the ratio between the window size and the standard size 64×128. Since windows with height smaller than 32 pixels or larger than H are not suitable for detection, we set the feasible range for $winsize$ to be $[0.25, H/128]$.

3.3.2. Definition of fitness

For the fitness function of DE, the HOG-SVM output function given in (1) can be used directly. Each time the algorithm evaluates a new solution, the extraction of HOG feature is executed at runtime, and the results are fed to a trained linear SVM. Output that is larger than the pre-defined threshold gives a positive value of the function, which means that the objective region is found. In this sense, during the search process, individuals with larger fitness value should be more desired.

3.3.3. Evolutionary process

The DE algorithm for the search problem in human detection has the same framework of traditional DE. At the beginning of the algorithm, the j-th variable of the i-th individual is initialized as

$$x_{i,j}^0 = x_{min,j} + randnum(0,1) \cdot (x_{max,j} - x_{min,j}) \quad (2)$$

where $1 \le j \le 3$, $x_{min,j}$ and $x_{max,j}$ are the lower and the upper bound defined in Section 3.3.1, and randnum(0,1) is a random number on the range [0,1].

After initialization, the algorithm enters an evolutionary process composed of mutation, crossover, evaluation and selection operations iteratively.

Mutation: The mutation operator is applied to generate a mutant v_i^g based on each individual x_i^g, where g is the number of generations. The generation of mutant is based on a base individual learning from the difference of other individuals. Depending on the choice of base individual and the individuals to learn from, there are several different mutation strategies. In this work, we adopt the DE/target-to-best/1, which can be expressed as

$$v_i^g = x_i^g + F \cdot (x_{best}^g - x_i^g) + F \cdot (x_{r1}^g - x_{r2}^g) \quad (3)$$

where the indices $r1$ and $r2$ are distinct integers randomly generated within the range $[1, popsize]$ excluding the integer i, x_{best}^g denotes the individual with the best-so-far fitness. The coefficient F is a positive parameter for controlling the effect of the differences. Here DE/target-to-best/1 is selected because the strategy draws individuals to the best-so-far individual and at the same time learns from the differences from other individuals.

Crossover: The crossover operator exchanges a number of variables of the mutant v_i^g with the x_i^g to reproduce a candidate u_i^g for the next generation. The process is controlled by a parameter CR, which can be expressed as

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{if } randnum(0,1) \le CR \quad \text{or} \quad j = j_{rnd} \\ x_{i,j}^g & \text{otherwise} \end{cases} \quad (4)$$

where j_{rnd} is an integer randomly generated from the range [1, 3] for guaranteeing that at least one variable is changed, and $randnum(0,1)$ is a randomly generated real number within the range [0,1]. The parameter CR is the crossover probability which controls the probability for variables to be inherited from the mutant.

Selection: After the evaluation of the trial vectors, the selection operation is finally performed to decide whether the reproduced trial vectors can survive to the next generation. For the search problem addressed in this work, a trial vector survives if its fitness is better than the corresponding vector, which can be expressed as

$$x_i^{g+1} = \begin{cases} u_i^g & \text{if } f(u_i^g) \ge f(x_i^g) \\ x_i^g & \text{otherwise} \end{cases} \quad (5)$$

where $f(x)$ is the fitness function defined in Section 3.3.2.

3.3.4. Termination criterion

The DE algorithm terminates when a pre-defined criterion is satisfied. In this work, the termination criterion is defined by a parameter FE_{max}, which denotes the maximum number of function evaluations (FEs). Since the computation of HOG-SVM output function is the most time-consuming part of the proposed method, the FE_{max} is directly related to the detection time.

3.3.5. Computational complexity

As introduced above, the most time-consuming part of HOG-SVM-DE is the evaluation of HOG-SVM output function. Compared with time consumption in the extraction of 3780 features, the time consumption of evolutionary operations in

DE would be trivial. The DE algorithm terminates after FE_{max} evaluations of HOG-SVM output function. Here the FE_{max} is set by the user. The time consumption of HOG-SVM-DE depends on the setting of FE_{max}. However, it is unwise to reduce the value of FE_{max} to a minimum, for large values of FE_{max} are excepted to allow more thorough search, and consequently promise better solution.

Since DE is a metaheuristic algorithm and the algorithm behavior is not known *a priori*, it is hard to discuss the effect of FE_{max} without experiments. In Fig. 5 an example is presented to show different values of FE_{max} and the corresponding objective window found by the proposed HOG-SVM-DE. The number of possible detection windows in the image is defined to be 2400. It is observed that the proposed HOG-SVM-DE checked only about 200 detection windows to find the objective window, which is more than 10 times faster than scanning these detection windows one by one. The improvement is significant, although the HOG-SVM-DE algorithm cannot promise the same results in every single run.

In this work, we would conduct experiments on more test images to further investigate the effect of FE_{max} in Section 4.

4. Experiments and discussions

4.1. Experimental setup

To study the performance of the proposed algorithm, experiments are conducted on two datasets of images. For the training stage of linear SVM, both positive and negative samples are from the INRIA Person Dataset [27]. For experiments on the detection stage, two set of images, denoted as Dataset1 and Dataset2, are used. Dataset1 includes 150 positive images and 150 negative ones from the INRIA dataset, all of the images are 320. For Dataset2, we collected positive images larger than 640×480 from internet, and all of the images collected are with large areas of background. It is supposed that these images are more difficult for locating the objective window. The negative images are background images with high resolution from the NICTA Dataset [28]. Examples of positive and negative images from Dataset2 are presented in Fig. 6. All human objects on the images are with height larger than 32 pixels.

The proposed HOG-SVM-DE is compared to both the conventional dense grid and the Gaussian PSO approach in [13]. The setting of parameters in DE and PSO are as follows. For DE, the factor F and the crossover rate CR are set to $F=0.5$ and $CR=0.9$, and the population size *popsize* is set to 30. For Gaussian PSO, the population size is set to 20. All of the three algorithms stop immediately when a human is detected in the image or the FEs exceed FE_{max}.

All of the experiments are implemented under the environment of Intel Duo Core CPU@2.6 GHz, Visual studio 2010, Windows 7. The OpenCV functions are used for optimized implementation of HOG feature extraction.

4.2. Results and discussions

Experimental results are presented in Table 2, where the average execution time and detection rate are reported. Here the detection rate is defined as the percentage of

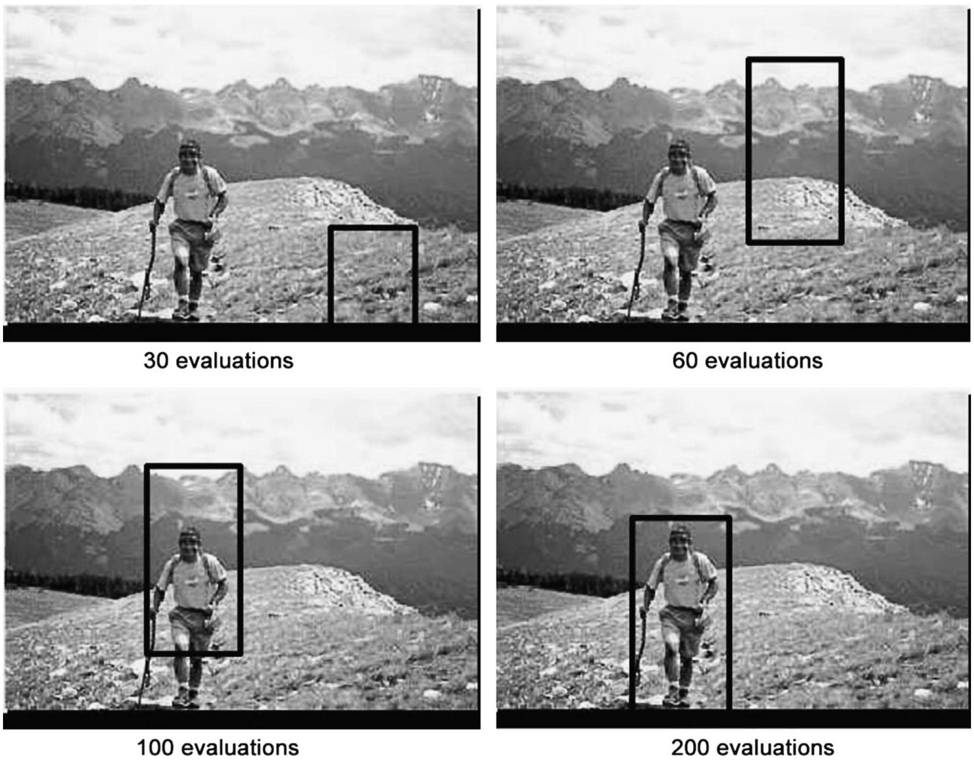

Fig. 5. HOG-SVM-DE running for 30, 60, 100, and 200 evaluations on a sample image.

images that are correctly labelled by the algorithm. The column FE_{max} denotes the maximum number of function evaluation for the PSO and the DE algorithm, and the columns s and l represent the step size for the position of sliding window and the number of different sizes in the traditional dense grid method, respectively.

It is observed from Table 2 that the proposed HOG-SVM-DE and the PSO approach in [12] consumes less than 200 ms on Dataset1, and less than 350 ms on Dataset2, while the traditional dense grid consumes much more time for $s=8$, $l=8$. Although the detection rates of HOG-SVM-DE and the PSO approach are less than that of the dense grid scanning, the two algorithms are acceptable for real-time applications for the advantage on computation time.

Note that when using the parameter set $s=16$ and $l=4$, the execution time of the dense grid method decreased to 296 ms for Dataset1 and 449 ms for Dataset2, but the detection rate of the dense grid method dropped drastically. Since the number of detection windows scanned has decreased, the drop of both execution time and detection rate is natural. In this case, the proposed HOG-SVM-DE still has significant advantage over the dense grid method in terms of detection rate.

When compared with the HOG-SVM-PSO, the proposed HOG-SVM-DE has shown advantage in both detection rate and execution time on the two datasets. The advantage is significant on Dataset2, where images with high resolution and large background are collected. In Fig. 7, cases where HOG-SVM-PSO failed and HOG-SVM-DE succeeded are presented. Both of the two cases are images with large areas of background and only one people, which requires efficient search algorithm to locate the objective detection window. On these cases, the proposed HOG-SVM-DE outperformed the PSO method, which is possibly resulted from the fast and global convergence of DE.

4.3. Parameter analysis

Generally, for the dense grid scanning method, the parameters s and l are directly associated with the time complexity and the detection rate of the algorithm. The number of windows scanned on average is in proportion to $W/s \times H/s \times l$, where W and H are the height and the width of the image, respectively. As verified in Table 2, using $2s$ and $(1/2)l$ would consequently decrease the execution time to about 1/8, but a sharp drop of accuracy would occur.

On the other hand, it is difficult to discuss the effect of different parameter settings in the proposed HOG-SVM-DE analytically, for DE is a meta-heuristic algorithm and the behavior is not known *a priori*.

Instead of analytical discussion, we investigate the relation between the maximum function evaluations, i.e. FE_{max} and the detection rate based on experiments. Here the maximum function evaluations are associated with the time consumption of HOG-SVM-DE, and the detection rate measures the successful rate of the algorithm. The results for HOG-SVM-DE running for a maximum of 400, 600, 800, and 1000 FE_{max} are reported in Table 3. It is observed that the relationship between the FE_{max} and the execution time is almost linear, while the relationship between FE_{max} and detection rate is like a convex curve. Based on these

Table 2
Experimental results for dense grid searching algorithm, HOG-SVM-PSO and HOG-SVM-DE.

Data	Algorithm	s	l	FEs_max	Detection rate (%)	Exe. time (ms)
Dataset1	DenseGrid	8	8	/	97	1882
Dataset1	HOG-SVM-PSO	/	/	400	80	189
Dataset1	HOG-SVM-DE	/	/	400	84	165
Dataset1	DenseGrid	16	4	/	69	296
Dataset2	DenseGrid	8	8	/	91	5673
Dataset2	HOG-SVM-PSO	/	/	600	73	311
Dataset2	HOG-SVM-DE	/	/	600	83	265
Dataset2	DenseGrid	16	4	/	47	449

Fig. 6. Sample images from Dataset2. (a) Positive images. (b) Negative images.

Fig. 7. Detection results for dense grid scanning, HOG-SVM-PSO and HOG-SVM-DE on two sample images.

Table 3
Experimental results for HOG-SVM-DE with different maximum numbers of FEs.

Dataset	FEs	Detection rate	Execution time (ms)
Dataset1	400	84	165
Dataset1	600	86	248
Dataset1	800	91	312
Dataset1	1000	92	409
Dataset2	400	76	177
Dataset2	600	83	265
Dataset2	800	85	332
Dataset2	1000	86	423

results, a maximum of 400 and 600 FEs are recommended for Dataset1 and Dataset2, respectively.

Given the empirical studies, the balance between detection rate and execution time of HOG-SVM-DE can be tuned with different settings of maximum FEs to satisfy different applications.

5. Conclusions

In this work, a fast human detection approach based on HOG-SVM with differential evolution (HOG-SVM-DE) is proposed. The proposed method applied the evolutionary computation technology differential evolution (DE) to locate the objective detection window. The DE algorithm works at the real-time detection stage and solves the non-linear search problem of locating the objective detection window. Individuals in DE are defined as detection windows on the image, and the fitness of individuals is assigned according to the output of HOG-SVM for the corresponding detection window. Experimental results verified that the proposed method is promising by achieving both favorable time efficiency and acceptable detection rate.

Future work would include extending the proposed algorithm to detect multiple people by using a multimodal DE algorithm. Besides, information-reusing strategies would be adopted in the proposed frame work to accelerate the computing of HOG feature. Applications on human tracking would also be considered.

References

[1] C. Papageorgiou, T. Poggio, A trainable system for object detection, Int. J. Comput. Vis. 38 (1) (2000) 15–33.
[2] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, Int. J. Comput. Vis. 77 (2008) 259–289.
[3] D. Gavrila, A bayesian, exemplar-based approach to hierarchical shape matching, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1408–1421.
[4] W. Gao, H. Ai, S. Lao, Adaptive contour features in oriented granular space for human detection and segmentation, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1786–1793.
[5] Y. Liu, S. Shan, W. Zhang, X. Chen, W. Gao, Granularity-tunable gradients partition descriptors for human detection, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1255–1262.
[6] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[7] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell 34 (4) (2012) 743–761.
[8] Z. Lin, L. Davis, A pose-invariant descriptor for human detection and segmentation, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 423–436.
[9] W. Zhang, G. Zelinsky, D. Samaras, Real-time accurate object detection using multiple resolutions, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.
[10] P. Dollar, B. Babenko, S. Belongie, P. Perona, Z. Tu, Multiple component learning for object detection, in: Proceedings of 10th European Conference on Computer Vision, 2008, pp. 211–224.
[11] J. Zhang, K. Huang, Y. Yu, T. Tan, Boosted local structured hog-lbp for object localization, in: Proceedings of the IEEE International Conference on Computer Vision Pattern Recognition, 2011, pp. 1393–1400.
[12] Q. Zhu, M.-C. Yeh, K.-T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, in: IEEE Conference on Computer Vision Pattern Recognition, vol. 2006, 2006, pp. 1491–1498.
[13] S.-T. An, J.-J. Kim, J.-W. Lee, J.-J. Lee, Fast human detection using gaussian particle swarm optimization, in: IEEE International Conference on Digital Ecosystems and Technologies, vol. 2011, 2011, pp. 143–146.
[14] Y. Pang, Y. Yuanb, X. Lib, J. Pan, Efficient hog human detection, Signal Process. 91 (4) (2011) 773–781.
[15] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, N. Navab, Dominant orientation templates for real-time detection of texture-less objects, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), 2010.
[16] C. Hong, J. Zhu, M. Song, Y. Wang, Realtime object matching with robust dominant orientation templates, in: International Conference on Pattern Recognition (ICPR2012), 2012, pp. 1152–1155.

[17] R.M. Storn, K.V. Price, Differential Evolution—a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces [Online]. Available: ⟨http://icsi.berkeley.edu/∼storn/litera.html⟩ [accessed: April 10, 2014], 1995, pp. 95–012.
[18] R.M. Storn, K.V. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[19] W.-J. Yu, M. Shen, W.-N. Chen, Z.-H. Zhan, Y.-J. Gong, Y. Lin, O. Liu, J. Zhang, Differential evolution with two-level parameter adaptation, IEEE Trans. Cybern. 44 (7) (2014) 1080–1099.
[20] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[21] M. Wu, B. Scholkopf, Transductive classification via local learning regularization, in: Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS), 2007, pp. 628–635.
[22] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (7) (2012) 3262–3272.
[23] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.
[24] J. Yu, Y. Rui, Y.Y. Tang, D. Tao, High-Order Distance-Based Multiview Stochastic Learning in Image Classification. Cybernetics, IEEE Transactions, PP (99) (2014) 1, http://dx.doi.org/10.1109/TCYB.2014.2307862.
[25] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H.S.-H. Chung, Y. Li, Y.-H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2013) 241–258.
[26] W.-N. Chen, J. Zhang, Ant colony optimization for software project scheduling and staffing with an event-based scheduler, IEEE Trans. Softw. Eng. 39 (1) (2013) 1–17.
[27] INRIA Person Dataset, [Online]. Available: ⟨http://pascal.inrialpes.fr/data/human⟩ [accessed: April 10, 2014].
[28] NICTA Pedestrian Dataset, [Online]. Available: ⟨http://nicta.com.au/research/projects/automap⟩ [accessed: April 10, 2014].